

3 associating a process being created with the virtual process identifier of a
4 process that made the system call.

1 6. The method of claim 5 further comprising:
2 storing object code comprising instructions to associate the process being
3 created with the virtual process identifier of the process that made
4 the system call; and
5 wherein intercepting comprises replacing a pointer to the system call with
6 a pointer to the stored object code, such that calling the system call
7 causes the object code to execute.

1 7. The method of claim 6 wherein storing object code comprises inserting the
2 object code into the operating system.

1 8. The method of claim 7 wherein:
2 inserting object code into the operating system comprises loading a
3 module into a running operating system kernel, the module
4 comprising the object code.

1 9. The method of claim 1 further comprising:
2 loading the first process into computer memory by a modified loader
3 program;
4 starting the first process by the modified loader program; and

004040-287960

5 storing, by the modified loader program, an entry in a data structure in
6 computer memory, the entry comprising a virtual process identifier
7 and a process identification number of the first process.

1 10. The method of claim 1 further comprising:

2 starting the first process by a virtual process manager program; and
3 storing, by the manager program, an entry in a data structure in computer
4 memory, the entry comprising a virtual process identifier and a
5 process identification number of the first process.

1 11. A computer program product for associating identifiers with virtual

2 processes, the computer program product comprising:

3 program code for starting, for each virtual process, a separate first process;
4 program code for associating each first process with a separate virtual
5 process identifier;

6 program code for originating, from each virtual process, additional

7 processes included in the virtual process;

8 program code for associating all processes that originate from each first

9 process with the associated virtual process identifier; and

10 a computer readable medium on which the program codes are stored.

1 12. The computer program product of claim 11 further comprising:

2 program code for storing an entry in a data structure in computer memory,

3 the entry comprising a virtual process identifier and a process

4 identification number of the first process.

1 13. The computer program product of claim 11 further comprising:
2 program code for intercepting system calls that create processes; and
3 program code for associating a process being created with the virtual
4 process identifier of a process that made the system call.

1 14. The computer program product of claim 11 further comprising:
2 program code comprising instructions to associate the process being
3 created with the virtual process identifier of the process that made
4 the system call; and
5 program code for replacing a pointer to the system call with a pointer to
6 the instructions to associate, such that calling the system call
7 causes the instructions to associate to execute.

1 15. The computer program product of claim 14 further comprising:
2 program code for storing the instructions to associate in the operating
3 system by loading a module into a running operating system
4 kernel, the module comprising the instructions to associate.

1 16. The computer program product of claim 11 further comprising:
2 program code for starting the first process; and
3 program code for storing an entry in a data structure in computer memory,
4 the entry comprising a virtual process identifier and a process
5 identification number of the first process.

4 the association module is further for associating a process being created
5 with the virtual process identifier of a process that made the system
6 call.

1 21. The system of claim 20 further comprising:
2 a storage module, coupled to the interception module, for storing object
3 code comprising instructions to associate the process being created
4 with the virtual process identifier of the process that made the
5 system call; and wherein
6 the interception module is further for replacing a pointer to the system call
7 with a pointer to the stored object code, such that calling the
8 system call causes the object code to execute.

1 22. The system of claim 21 further comprising:
2 coupled to the storage module, an insertion module, for inserting the
3 instructions to associate in the operating system by loading a
4 module into a running operating system kernel.

1 23. A method in a computer system for associating identifiers with virtual
2 processes, the method comprising:
3 starting each virtual process by executing a separate, system initialization
4 process;
5 associating each system initialization process with a virtual process
6 identifier;

